(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 April 2002 (25.04.2002)

PCT

(10) International Publication Number
WO 02/33661 A1

(51) International Patent Classification⁷: G06T 17/00

(21) International Application Number: PCT/US01/32246

(22) International Filing Date: 17 October 2001 (17.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/691,574 17 October 2000 (17.10.2000) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/691,574 (CON)
Filed on 17 October 2000 (17.10.2000)

(71) Applicant *(for all designated States except US)*: **ACTUALITY SYSTEMS, INC.** [US/US]; 315 Main Street, Suite 102, Reading, MA 01867 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **BAXTER, William, V., III** [US/US]; 500 Woodcroft Parkway, Durham, NC 27713 (US). **GIOVINCO, Michael** [US/US]; 12 Ivy Street, Cambridge, MA 02138 (US).

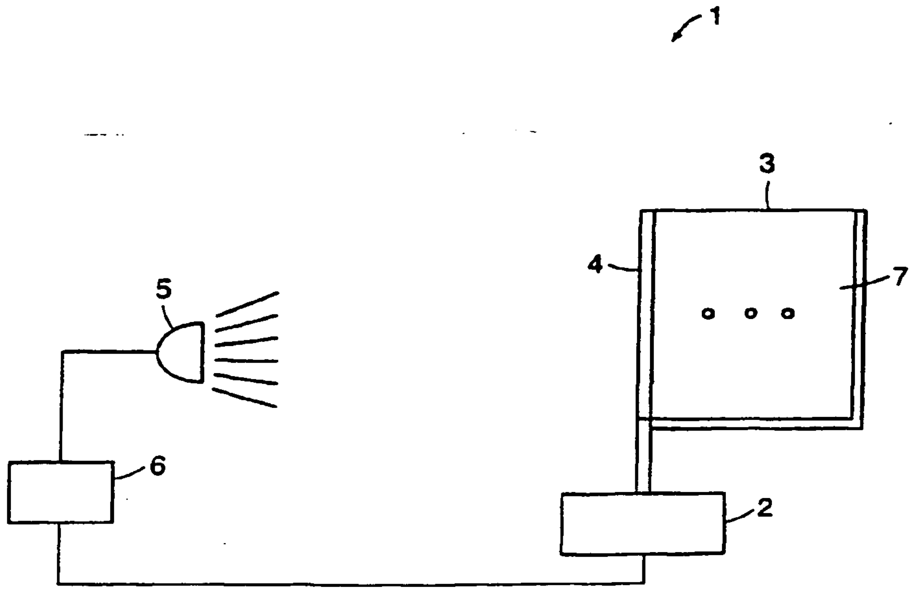
(74) Agent: **PRAHL, Eric**; Fish & Richardson P.C., 225 Franklin Street, Boston, MA 02110 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,

[Continued on next page]

(54) Title: RASTERIZATION OF LINES IN A CYLINDRICAL VOXEL GRID



(57) Abstract: In a volumetric display (1), (Fig.1), a motor (2) spins an imaging plate (3) rapdily about an axis (4). A light source (5) under the control of a rasterizer (6) illuminates selected spots (7) on the imaging plate (3) at sucessive instants. If the imaging plate (3) spins rapidly enough, and the successive instants are separeted by short enough time intervals, a continuous curve will appear to hang in mid-air. A rasterizer renders a desired line on a volumetric display. It does so by determining whether first and second points on the desired line are on first and second constituent line-segments of the desired line, the first and second constituent line-segments having different dominant directions.

CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

RASTERIZATION OF LINES IN A CYLINDRICAL VOXEL GRID

This invention relates to computer graphics, and in particular, to the rendering of a straight line on a display in which the voxels are arranged on a cylindrical coordinate grid.

BACKGROUND

Many computer-implemented displays consist of two-dimensional arrays of individual picture elements, or pixels. To form an image, a rasterizer selectively colors the pixels. Because the individual pixels are so small, the display appears to a human viewer to be a continuous rendering of an image. This illusion is particularly effective for complex images of continuous tones such as photographs.

For simple geometric shapes, however, the pixelated nature of the display can become apparent to the human viewer. For example, if the rasterizer is instructed to draw a straight line, there is no guarantee that the points on that desired line will coincide with the pixels that are available for rendering the line. As a result, the desired line is often rendered by a rasterized line of pixels that are close to but not necessarily coincident with the desired line. This results in rasterized lines that have a jagged or echeloned appearance.

In the course of rendering an image, a large number of straight lines and line segments are often drawn. As a result, given a desired line, the rasterizer must frequently select those pixels that will minimize the jagged appearance of the resulting rasterized line. A straightforward mathematical approach is to use the equation of the desired line and the coordinates of the available pixels to minimize a least square error across all points on the line. While such an approach has the advantage of globally optimizing the selection of pixels on the rasterized line, the large number of floating point operations required causes this approach to be prohibitively time-consuming.

To meet constraints on speed, rasterizers typically implement rasterization methods that avoid time-consuming floating point operations. Among the methods that meet the foregoing constraints is that taught in *Bresenham, J.E.*, "Algorithm for Computer Control of a Digital Plotter," IBM System Journal, Vol. 4, pp.25-30, 1965, the contents of which are herein incorporated by reference. Using only integer operations, the Bresenham algorithm reduces the choice of what pixel to select to an examination of the sign of a discriminant.

Even faster rasterization methods exist that select multiple points on the rasterized line based on the outcome of a single decision. There also exist a variety of rasterization methods aimed at rendering conic sections and quadric sections on a pixelated display.

However, the foregoing rasterization methods all rely on the assumption that the array of pixels is arranged in a uniform rectangular grid that can readily be modeled by a Cartesian coordinate system. This is a reasonable assumption given the prevalence of two-dimensional displays such as computer monitors and printers at the time these algorithms were developed.

Since then, however, volumetric, or three-dimensional displays have been developed. Such displays permit the generation, absorption, or scattering of visible radiation from a set of localized and specified regions within a volume. Examples of such systems are taught in *Hirsch* U.S. Patent No. 2,967,905, *Ketchpel* U.S. Patent No. 3,140,415, *Tsao* U.S. Patent No. 5,754,147 and on pages 66-67 of *Aviation Week*, October 31, 1960.

In a typical volumetric display **1**, shown in FIG. 1, a motor **2** spins an imaging plate **3** rapidly about an axis **4**. A light source **5** under the control of a rasterizer **6** illuminates selected spots **7** on the imaging plate **3** at successive instants. If the imaging plate **3** spins rapidly enough, and if the successive instants are separated by short enough time intervals, a continuous curve will appear to hang in mid-air.

FIG. 2 illustrates the projection **8** onto a plane perpendicular to the axis **4** of an exemplary line rendered by the rasterizer **6**. In FIG. 2, the imaging plate **3** is shown in six successive instants as it rotates around the axis **4**, now perpendicular to the page. At each of the six instants, the light source **5**, under the control of the rasterizer **6**, illuminates a spot **7** on the imaging plate **3**. As shown in FIG. 2, by aiming the light source **5** at the correct spot and firing the light source **5** at the right time, it is possible to trace out the line. It is the function of the rasterizer **6** to correctly aim and fire the light source **5** so as to trace out the line.

To aim the light source **5**, the rasterizer **6** needs a way to identify points in space. In other words, the rasterizer **6** needs a coordinate system. One possible coordinate system is a Cartesian coordinate system.

Using a Cartesian coordinate system, the rasterizer **6** would draw the line **8** in FIG. 2 by specifying, for example, that at time t_1, the light-source **5** should aim 30 units north, at time t_2, the light-source **5** should aim 29 units north, 1 unit east and 1 unit in elevation, at time t_3, the light-source **5** should aim 28 units north, 2 units east and 2 units in elevation, and so on. To ensure that the imaging plate **3** is at the appropriate location, the rasterizer **6** also has to specify the values of the times t_1, t_2 and t_3 based on how fast the imaging plate **3** is spinning. Although the spinning of the imaging plate **3** can be resolved into a north-south component and an east-west component, this is a computationally taxing exercise that can easily be avoided by using a polar coordinate system.

As an alternative, the rasterizer **6** could draw the line in FIG. 2 in cylindrical coordinates by specifying, for example, that at time t_1, the light source **5** should aim 30 units away from the axis **4** at a 90 degree angle, at time t_2, the light source **5** should aim 29.02 units from the axis **4** at an angle of 88.03 degrees and point up by 1 unit of elevation, and at time t_3, the light source **5** should aim at 28.07 units from the axis **4** at an angle of 85.91 degrees and point up by 2 units of elevation. This would, of course, draw the same line that was drawn using Cartesian coordinates. The advantage of using cylindrical coordinates becomes apparent when it comes time to specify when to fire the light source **5**. Because the imaging plate **3** is spinning, it is far more natural to represent its motion in terms of degrees per second than it is to resolve its motion into a north-south component and an east-west component. For this reason, calculations involving motion of the imaging plate **3** in a volumetric display are best performed in terms of a cylindrical coordinate system.

The process of rendering a line in a cylindrical coordinate system is fundamentally different from that of rendering a line in a Cartesian coordinate system. In a Cartesian coordinate system, a line has a constant slope. A unit change in one direction always results in the same change in the remaining two directions, regardless of where that change occurs. This is not the case in a cylindrical coordinate system.

In a cylindrical coordinate system, the slope of a line can vary dramatically with position along the line. In contrast to a Cartesian coordinate system in which the slope of a line is constant at all points on the line, the slope of a line in a cylindrical coordinate system can vary significantly along the line. This is because the coordinate grid in a cylindrical coordinate system is not a spatially uniform grid, as is the coordinate grid in a Cartesian coordinate system.

For those portions of the line that are closest to the axis of the cylindrical coordinate system, the change in distance of the line from the origin changes very slowly with angle. For those portions of the line that are far from the axis of the cylindrical coordinate system, even a small change in angle results in a huge change in radial distance from the axis.

SUMMARY

A rasterizer according to the invention recognizes that the dominant direction of a desired line in a cylindrical coordinate system changes in a predictable manner. The rasterizer uses this predictable sequence of dominant directions, referred to as a "dominance progression," to avoid having to determine the dominant direction at each point on the desired line before rendering a voxel representative of that point. Instead, the rasterizer detects boundaries between constituent line-segments of the desired line and uses the dominance progression to determine the dominant direction of the line at any point. Because the process of detecting a transition between boundaries is much faster than that of evaluating a dominant direction, a rasterizer practicing the method of the invention can rapidly and efficiently draw a rasterized line in a cylindrical coordinate system.

The rasterizer renders, on a volumetric display, a rasterized line that approximates a desired line. It does so by rendering, on the volumetric display, a first voxel representative of a first point from the desired line. The first point is selected from a first constituent line-segment having a first dominant direction. Then, the rasterizer selects a second point from the desired line and determines whether that second point is on a second constituent line-segment, with a second dominant direction, or whether that second point is on the same constituent line segment.

On the basis of this determination, the rasterizer renders a second voxel representative of the second point on the desired line. This second voxel is located so as to minimize a distance between the second voxel and the second point on the desired line. The second voxel can be selected, for example, by application of a midpoint method to select the second voxel from a plurality of candidate voxels.

In one practice of the invention, the rasterizer evaluates a difference between a threshold and a transition-indicating expression evaluated at the second point. This difference is indicative of a transition from the first constituent line-segment to the second constituent line-segment. The transition indicating-expression is typically an expression whose value depends on a slope of the desired line at the second point. The slope can be defined as a ratio of how much the desired line changes in the first dominant direction with respect to a change in one of two non-dominant directions.

In some cases, the orientation of the line to be rasterized is such that the transition cannot be reliably detected. This occurs, in part, because of computation on a digital computer cannot be carried out with infinite precision. In one practice of the invention, erroneous transitions are avoided by selecting the transition-indicating expression to be one that identifies a line having a point at which the slope is equal to the threshold.

Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. Although methods and materials similar or equivalent to those described herein can be used in the practice or testing of the present invention, suitable methods and materials are described below. All publications, patent applications, patents, and other references mentioned herein are incorporated by reference in their entirety. In case of conflict, the present specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

Other features and advantages of the invention will be apparent from the following detailed description, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a volumetric display for practice of the rasterization method of the invention;

FIG. 2 is a projection of a desired line on the r-θ plane;

FIGS. 3-4 show two views of the desired line and a cylindrical surface to which it is tangent;

FIGS. 5, 6, 7A and 7B show curves derived from three discriminant functions used in rasterizing the desired line;

FIG. 8 is a flow chart summarizing the steps in the practice of the invention;

FIGS. 9-11 show curves derived from the three discriminant functions of FIGS. 6, 6, and 7A together with the locus of points for which their derivatives are equal to one;

FIGS. 12-13 are examples of cases in which the proximity of a discriminant function to the locus of points in FIGS. 10 and 11 gives rise to numerical instability;

DETAILED DESCRIPTION

A line traversing a cylindrical coordinate system defines a sequence of points, each of which is defined by a radial coordinate r, an angular-coordinate θ, and an axial coordinate h. As one traverses the line, these points change by different amounts depending on the direction of the line. For example, if the line is nearly parallel to the axis of the cylindrical coordinate system (hereafter referred to as the "axis"), r and θ will change by only small amounts compared to the change in h. Such a line is referred to as "h-dominant" because its dominant direction is the axial direction. If the line passes through the axis of the cylinder and deviates only slightly from a plane orthogonal to that axis, then θ will never change at all and h will change by only a small amount. Such a line is referred to as "r-dominant" because its dominant direction is the radial direction. For short lines that do not pass through the axis, θ may change more rapidly than either r or h. Such lines are said to be "θ-dominant" because their dominant direction is the angular direction.

As a general rule, the dominant direction of a line in a cylindrical coordinate system changes as one traverses the line. It is useful therefore to consider a line as a concatenation of line-segments, each of which has a dominant direction that differs from its neighboring line-segments. Within each line-segment, the dominant direction of the line is the same. However, when crossing the boundary between two line-segments, the dominant direction changes.

The geometric meaning of "line" as a set of points extending infinitely in both directions is not useful in a discussion of a volumetric display having finite extent. Hence, throughout this specification, the word "line" generally refers to line-segments defined by two endpoints. With this definition, it becomes meaningful to refer to lines as being "short" or "long." A "short line" is a line-segment whose length is short relative to the extent of the volumetric display. A "long line" is a line-segment other than a short line.

In the rasterization method of the invention, the rasterizer **6** begins at one endpoint of the line and identifies the dominant direction of the line-segment containing that endpoint. To reach the next point on the line, the rasterizer **6** first increments the value of the coordinate (hereafter referred to as the "dominant coordinate) associated with the dominant direction for that line-segment. Then, the rasterizer **6** decides whether to increment the values of coordinates associated with one or the other non-dominant direction.

To determine whether to increment the values of the coordinates associated with the non-dominant directions, it is useful to define three discriminant functions, each of which defines a surface in the cylindrical coordinate system. Each of the three coordinates is an argument to two of these discriminant functions. The discriminant functions are selected so that the intersection of any two of them defines the desired line. For convenience of notation, these discriminant functions are referred to as:

$$F(r,\theta)=0$$

$$G(r,h)=0$$

$$H(\theta,h)=0$$

The functional forms of the above three discriminant functions are best understood with reference to FIGS. 3 and 4. As shown in those figures, a desired line **10** in cylindrical coordinates can be conveniently described in terms of four geometric measurements. A tangent plane containing the desired line **10** is defined by the line's **10** point of tangency **14** (r_t,α,h_t) with a cylindrical surface **15**. The specific line within this tangent plane is defined by the angle of twist, β, of the desired line **10** in the tangent plane.

The first discriminant function, $F(r,\theta)$, is the projection of the desired line **10** onto a plane that is perpendicular to the axis of the cylindrical coordinate system:

$$F(r,\theta)=r\cos(\theta-\alpha)-r_t=0$$

where r_t is the radial distance to the point of tangency **14** and α is the angular coordinate at the point of tangency **14**. An intersection of the r-θ plane and the surface defined by $F(r,\theta)=0$ is shown in FIG. 5.

The second discriminant function, $G(r,h)$, is a positive branch of a hyperbola. The hyperbola has an axis of symmetry extending in the h direction and passing through the point of tangency **14** at (r_t, h_t). The angle formed by the r-axis and the hyperbola's asymptotes is β. The hyperbola is derived from the desired line **10** using the equation:

$$G(r,h)=\eta(r^2-r_t^2)-(h-h_t)^2=0$$

where η is defined as $\tan^2(\beta)$. FIG. 6 shows the curve satisfying the constraint $G(r,h)=0$ in the r-h parametric domain.

The third discriminant function, $H(h,\theta)$, defines a tangent function in the θ-h *parametric domain.* The inflection point of this tangent function is located at the point of tangency **14**, $(\theta=\alpha,h=h_t)$. The equation is given by:

$$H(h,\theta)=\xi\tan(\theta-\alpha)+h_t-h=0$$

where ξ is defined to be $r_t\tan(\beta)$. FIGS. 7A and 7B show curves defined by $H(h,\theta)=0$ in the θ-h parametric domain corresponding to positive and negative values of ξ.

Note that unlike the first two discriminant functions, the sign of the slope of $H(h,\theta)$ changes depending on the sign of ξ. In implementing the third discriminant function, the rasterizer **6** can use two separate (but very similar functions). Alternatively, the rasterizer **6** can incorporate the sign of ξ into a single function call. The use of two separate functions may be more efficient because doing so avoids extra operations associated with changing signs of subexpressions.

A dominant direction characterizes each point on the desired line **10**. The dominant direction associated with a point on the desired line is that direction in which the coordinate for that direction changes most rapidly relative to the other coordinates. The rasterizer **6** identifies this direction by using the above three functions to evaluate the derivatives dr/dh, $dr/d\theta$, and $dh/d\theta$. Because of the constraints $F(r,\theta)=G(r,h)=H(h,\theta)=0$, it follows that $dF=dG=dH=0$. Hence, one obtains the following expressions:

$$\frac{dr}{d\theta}=r\tan(\theta-\alpha)\frac{\text{radians}}{\text{step}}$$

$$\frac{dr}{dh}=\frac{h-h_t}{\eta r}$$

$$\frac{dh}{d\theta}=\xi\sec^2(\theta-\alpha)\frac{\text{radians}}{\text{step}}$$

To improve performance, angular coordinates are expressed as dimensionless integers rather than in radians. A "radians/step" constant for converting between dimensionless integers and radians is therefore incorporated within the θ argument of the trigonometric functions. The scale factor "radians/step" that appears in the first and third derivatives thus arises from the chain rule of elementary calculus.

Given a point on the desired line **10**, one can evaluate the three discriminant functions $F(r,\theta)$, $G(r,h)$ and $H(h,\theta)$. The derivatives of these three functions at that point on the desired line **10** identify the dominant direction at that point. Once the dominant direction is known, there exist known midpoint methods for determining successive voxels, thereby generating a rasterized line that approximates the desired line **10**.

Because of the time required to evaluate the three discriminant functions and their derivatives at each point on the desired line **10**, it is preferable to instead predict the progression of dominance one encounters while progressing along the desired line **10** and to identify indicia of a change in the dominant direction of the desired line **10**. For a line in cylindrical coordinates, a dominance progression can be derived by observing the behavior of the three discriminant functions as θ proceeds from $\theta = \alpha - \pi/2$ to $\theta = \alpha + \pi/2$.

The dominance progression for the discriminant function $F(r,\theta)$ is that of a secant function. As θ increases, $F(r,\theta)$, and hence the desired line **10** from which it was derived, progresses from being *r*-dominant, to being *θ*-dominant, and then back to being *r*-dominant.

The dominance progression for the discriminant function $G(r,h)$ is that of one branch of a hyperbola. In this case, the dominance progression depends on the slopes of the asymptotes. If the slopes of the asymptotes have a magnitude greater than one, then $G(r,h)$, and hence the desired line **10** from which it was derived, progresses from being *r*-dominant, to being *h*-dominant, and then back to being *r*-dominant, switching sign at the point of tangency **14** (h_t, r_t). However, if the slopes of the asymptotes have magnitudes that are less than one, the hyperbola never becomes *r*-dominant. Under these circumstances, *h* is more dominant than *r* at all points on the desired line **10**. Geometrically this corresponds to a desired line **10** that is nearly parallel to the cylindrical axis.

The third discriminant function, the tangent function $H(h,\theta)$, generally progresses from being *h*-dominant, to being *θ*-dominant and finally back to being *h*-

dominant. However, the minimum slope of a tangent function is equal to its coefficient, ξ. Hence, if ξ is greater than one, then the slope of the tangent function $H(h,\theta)$ is everywhere greater than one. Thus it is possible for a line to be h-dominant everywhere and never θ-dominant. Geometrically this corresponds to a desired line **10** for which either r_t is large, or to a desired line **10** that is nearly parallel to the cylindrical axis.

The dominance progression for each of the three discriminant functions can therefore be summarized in the following table:

increasing $\theta \rightarrow$			
$F(r,\theta)$	r	θ	r
$G(r,h)$	(r)	h	(r)
$H(h,\theta)$	h	(θ)	h

In the above table, parentheses enclose those coordinates that, for certain lines, may never become dominant. Note that the inflection point on each of these three discriminant functions coincides with the line's point of tangency **14** (r_t,α,h_t) with the cylindrical surface **15**.

The middle column of the table indicates that at the point of tangency **14**, r can never be dominant. At that point, only θ or h can be dominant. Which of the two is dominant can be determined by observing the behavior of $H(h,\theta)$. The dominance progression for $H(h,\theta)$ indicates that if there exists a segment on the desired line **10** in which θ is more dominant than h, the point on that line at which $\theta = \alpha$ will be in that region. Consequently, θ dominance, if present at all, will manifest itself near the $\theta = \alpha$ point on the line.

Proceeding outward from the $\theta = \alpha$ point on the desired line **10**, the next coordinate to become dominant will be either r or h. The dominance progression of the hyperbola, $G(r,h)$ indicates which one it will be. Based on the dominance progression of $G(r,h)$, it is apparent that if there exists an h-dominant line-segment on the desired line **10**, that line-segment will be closer to the point of tangency **14** than any r-dominant segment.

Based on the foregoing, it is always the case that, in traversing a line in cylindrical coordinates from $\theta = \alpha - \pi/2$ to $\theta = \alpha + \pi/2$, the dominant directions of the consecutive line-segments that constitute the desired line **10** will progress in the following order:

$$r \rightarrow h \rightarrow \theta \rightarrow h \rightarrow r$$

Note that the above dominance progression does not require that each direction be dominant at some point on the desired line **10**. Instead, the above dominance progression states that if the desired line **10** includes a line-segment in which a particular direction dominates, then that line-segment will appear in relation to the other line-segments as indicated above. In referring to a line-segment, a "+" or "–" sign indicates whether that line-segment lies before the point of tangency **14** or after the point of tangency **14**. Thus the constituent line-segments of the desired line **10** are conveniently labeled as follows:

$$R^- \rightarrow H^- \rightarrow \theta^- \rightarrow \theta^+ \rightarrow H^+ \rightarrow R^=$$

The recognition of the foregoing dominance progression in any line in cylindrical coordinates enables the implementation of a rasterization method that does not require computationally expensive evaluation of the three discriminant functions to identify the dominant direction at each point on the line. Instead, rasterization proceeds following steps shown in FIG. 8.

The rasterization method shown in FIG. 8 begins with an initialization step in which the rasterizer **6** renders the first voxel (step **16**). This first voxel corresponds to a first endpoint of the desired line **10**. The first endpoint is selected so that traversing the desired line **10** from the first endpoint to the second endpoint results in a decreasing r coordinate followed by an increasing r coordinate.

The initialization step is followed by steps in which the rasterizer **6** proceeds from one constituent line-segment of the desired line **10** to the next. Within a constituent line-segment, the rasterizer **6** steps unconditionally in the dominant direction of the line-segment. It then determines whether or not to also step in the non-dominant directions. To do so, the rasterizer **6** evaluates a discriminant function (one of $F(r,\theta)$, $G(r,h)$, or $H(h,\theta)$) associated with each of the other two non-dominant directions. After rendering each voxel, the rasterizer **6** updates all the slope values for the discriminant functions and checks to see if the next point is within the same line-segment.

Following the initialization step, the rasterizer **6** renders voxels representative of points in the R^- line-segment of the desired line **10**, if any (step **18**). In doing so, the rasterizer **6** evaluates the signs of the first and second discriminant functions, $F(r,\theta)$ and $G(r,h)$, to determine whether to step in the θ and h directions respectively. Before rendering a voxel representative of a point on the desired line **10**, the rasterizer **6** verifies that that point is still in the R^- line-segment of the desired line **10**. If it is not, then the rasterizer **6** moves to the step of rendering voxels representative of points on the H^- line-segment of the desired line **10**, if any (step **20**).

Once the rasterizer **6** determines that it has crossed over into the H^- line-segment of the line, it evaluates the signs of the second and third discriminant functions, $G(r,h)$ and $H(h,\theta)$, to determine whether to step in the r and θ directions respectively. Before rendering a voxel representative of a point on the desired line **10**, the rasterizer **6** checks to see if that point is still in the H^- line-segment of the desired line **10**. If it is not, then the rasterizer **6** moves to the step of rendering voxels representative of points on the θ^- line-segment of the desired line **10**, if any (step **22**).

Once the rasterizer **6** determines that it has crossed over into the θ-line-segment of the line, it evaluates the signs of the first and third discriminant functions, $F(r,\theta)$ and $H(h,\theta)$, to determine where to step in the r and h directions respectively. The rasterizer **6** continues to operate in this manner as long as θ remains the dominant direction. Hence, the rasterizer **6** operates in this manner even after crossing over into the θ^+ line-segment of the desired line **10** (step **24**).

Once in the θ^+ line-segment of the desired line **10**, the rasterizer **6** checks to see if the next point to be drawn is still in the θ^+ line-segment of the desired line **10**. If it is not, then the rasterizer **6** moves to the steps of rendering voxels representing points on the H^+ line-segment of the line (step **26**), if any, and voxels representing points on the R^+ line-segment of the line (step **28**), if any. These steps are carried out in a manner analogous to the earlier steps of rendering voxels representing points in the H^- and R^- line-segments of the desired line **10**.

To locate the boundaries of the constituent line-segments of the desired line **10**, the rasterizer **6** determines the slope of the discriminant function with respect to the currently dominant variable. If this slope transitions from being less than one to being greater than one (or vice versa), the rasterizer **6** interprets this as indicating that it has now crossed into a different constituent line-segment of the desired line **10** and that the dominant direction has changed in accordance with the progression of dominance identified above.

FIG. 9 shows the curve $F(r,\theta) = 0$ (the solid curve) together with the locus of all points for which $dr/d\theta$ is equal to one (the dashed curve). As is apparent from the figure, there are only two points in r-θ space that are both on the discriminant function and for which $dr/d\theta$ is equal to one. These two points, which correspond to the intersections of the dashed curve and the solid curve in the figure, unambiguously identify the points at which the line transitions from one constituent line-segment to the next. FIGS. 10 and 11 show the curves $G(r,h) = 0$ and $H(h,\theta) = 0$ (solid curves) and the corresponding loci of points for which dr/dh and $dh/d\theta$ are equal to one. As shown in FIGS. 10 and 11, there are only two points in each case that are both on the discriminant function and for which the derivative is equal to one.

For each of the three cases above, there also exist points that almost, but do not quite satisfy the two conditions set forth for each case. These points correspond to those that are in the neighborhood of the intersection of the dashed and solid curves in FIGS. 9-11. One such set of erroneous points is illustrated by the shaded region in FIG. 10.

Because the only points that can be rendered on the display are those that fall on the cylindrical coordinate grid, it is possible that the rasterizer **6** will mistakenly consider one of the points in the set of erroneous points to be the actual intersection of the dashed and solid curves. This may result in the rasterizer **6** assuming that a transition from one constituent line-segment to the next has occurred when in fact, no such transition has occurred.

For a desired line **10** whose discriminant functions are as shown in FIGS. 9-11, a numerical error of the type described above will have little or no practical consequences. This is because in each of FIGS. 9-11, all points in the set of erroneous points are proximate to the true intersection point of the dashed and solid curves.

For some lines in a cylindrical coordinate system, however, the set of erroneous points can extend further away from the true intersection point of the dashed and solid curves. For example, referring back to FIG. 4, when the angle of twist β in the tangent plane approaches 45 degrees, $\tan(\beta)$ approaches one. When this occurs, the discriminant function $G(r,h)$ and the locus of points for which dr/dh is equal to one become close to each other even at points that are far away from their intersection. In this case, the set of erroneous points, shown in FIG. 12, extends far beyond the intersection of the two curves. Under these circumstances, an erroneous selection of one of these points can results in a grossly premature transition from one constituent line-segment to the next.

Another example of a desired line **10** that may result in a numerical error by the rasterizer **6** is one that passes very close to the cylindrical axis $(r_t \approx 0)$ or one in which the desired line **10** lies mostly in the r-theta plane $(\beta \approx 0)$. In either of these cases, ξ approaches zero. This causes the vertical dashed lines of FIG. 11 to move outward toward the tail of the tangent function, as shown in FIG. 13. Under these circumstances, the vertically extending set of erroneous points can include erroneous points are far away from the actual intersection of the dashed lines and the solid curve. This again can result in an inappropriate transition from one constituent line-segment to the next.

One way to eliminate numerical errors of the type described above is to symbolically compute the coordinates of the actual intersection of the dashed and solid curves in FIGS. 9-11 and to use those coordinates as transition thresholds. In practice it suffices to compute only one of the two coordinates that define the intersection. Of the two coordinates that define the intersection point, the more reliable is the dominant coordinate. For example, when in an *r*-dominant line-segment the rasterizer **6** would compare the current value of *r* on the desired line **10** with the *r* coordinate of the intersection to determine whether to transition out of the *r*-dominant line-segment. In terms of FIGS. 9-11, the symbolically computed coordinate defines either a horizontal or vertical line which goes through the intersection point.

Using the equations for $F(r,\theta)$ and $dr/d\theta$, one can eliminate a common variable *r* and derive the following expression for a vertical line in FIG. 9:

$$Fi(\theta) = \cos^2(\theta-\alpha) - r_t \sin(\theta-\alpha)\frac{\text{radians}}{\text{step}}$$

Similarly, using the equations for $G(r,h)$ and dr/dh, one can eliminate the common variable θ and derive the following equation for a horizontal line in FIG. 10:

$$Gi(r) = (1-\eta)r^2 - r_t^2$$

Finally, using the equations for $H(h,\theta)$ and $dH/d\theta$, one can again eliminate the common variable θ and derive the following expression for a horizontal line in FIG. 11:

$$Hi(h) = (h-h_t)^2 - (hSgn\cdot\xi\frac{\text{steps}}{\text{rad}} - \xi^2)$$

These three transition-indicating expressions are used as substitutes for expressions that are prone to numerical instability under the circumstances set forth above. For example, if the rasterizer **6** is in the R^- line-segment of a desired line **10**, it can recognize a transition to a successor line-segment upon recognition of the truth of either one of the following Boolean expressions:

$$\frac{dr}{d\theta} > -1\text{, or}$$

$$\frac{dr}{dh}\cdot hSgn > -1$$

However, the second expression is prone to numerical instability for certain lines. As a result, the Boolean expression

$$\frac{dr}{dh}\cdot hSgn > -1$$

is replaced by a Boolean expression that uses the more stable transition-indicating expression $Gi(r)$:

$$Gi(r) < 0.$$

The following table lists, for each constituent line-segment, a pair of Boolean expressions that can be used to indicate a transition from one constituent line-segment to another:

Constituent line segment	**Expression 1**	**Expression 2**
R^-	$\frac{dr}{d\theta} > -1$	$\frac{dr}{dh} \cdot hSgn > -1$
H^-	$\frac{dh}{d\theta} \cdot hSgn < 1$	$\frac{dr}{dh} \cdot hSgn < 0$
θ^-	$\frac{dr}{d\theta} > 0$	none
θ^+	$\frac{dr}{d\theta} \cdot hSgn > 1$	$\frac{dr}{d\theta} > 1$
H^+	$\frac{dr}{dh} \cdot hSgn > 1$	none
R^+	none	none

Certain of the foregoing Boolean expressions are prone to numerical instabilities. To avoid numerical instabilities, these Boolean expressions can be replaced by more stable Boolean expressions involving the transition-indicating expressions *Fi*, *Gi* and *Hi*. The following table lists alternative pairs of Boolean expressions for determining the occurrence of a transition between constituent line-segments. In the following table, all Boolean expressions that are prone to numerical instability have been replaced by more stable expressions involving the transition-indicating expressions *Fi*, *Gi* and *Hi*.

Constituent line segment	**Expression 1**	**Expression 2**

R^-	$\frac{dr}{d\theta} > -1$	$Gi < 0$
H^-	$Hi < 0$	$h \cdot hSgn > h_t \cdot hSgn$
θ^-	$\frac{dr}{d\theta} > 0$	none
θ^+	$\frac{dr}{d\theta} \cdot hSgn > 1$	$Fi < 0$
H^+	$Gi > 0$	none
R^+	none	none

The code provided as an appendix below shows one implementation of a rasterizer according to the invention. However, it is to be understood that while the invention has been described in conjunction with the detailed description thereof, the foregoing description and the accompanying code is intended to illustrate and not limit the scope of the invention.

APPENDIX

```
void writeLineFloatCylindrical(float x1, float y1, float z1,
                    float x2, float y2, float z2)
{
 // THE FOLLOWING STATIC CONST VALUES ARE FIXED FOR A GIVEN DEVICE
 static const float radsPerStep = M_2PI/g_stepsA;
 static const float stepsPerRad = 1.0/radsPerStep;
 static const float thetaStep = radsPerStep;
 static const float halfThetaStep = 0.5*radsPerStep;
 static const float thetaDomLimit =
   2*IntegerDoubleSin(1)/(5-4*IntegerDoubleCos(1));
 static const float MACHINE_EPSILON = 1e-7;
#ifdef INCREMENTAL_TRIG
 // COMPUTE VALUES FOR INCREMENTAL SIN/COS EVALUATION
 //   aW,bW for whole steps; aH,bH for half steps
 static const float sinQuarterThetaStep = sin(halfThetaStep*0.5);
 static const float aH = 2*sinQuarterThetaStep*sinQuarterThetaStep;
 static const float bH = IntegerDoubleSin(1);
 static const float aW = 2*bH*bH;
 static const float bW = IntegerDoubleSin(2);
#endif

#ifdef INCREMENTAL_TRIG
# define TRIG_HALF_STEP(c,s) \
  do { prevCos = c;  prevSin = s; \
     c = prevCos - (aH*prevCos + bH*prevSin);\
       s = prevSin - (aH*prevSin - bH*prevCos);  } while(0)
 # define TRIG_WHOLE_STEP(c,s) \
    do { prevCos = c;  prevSin = s; \
       c = prevCos - (aW*prevCos + bW*prevSin);\
       s = prevSin - (aW*prevSin - bW*prevCos);  } while(0)
#else
 # define TRIG_HALF_STEP(c,s) \
    do { c = cos( m_th*radsPerStep - tht ); \
       s = sin( m_th*radsPerStep - tht ); } while(0)
 # define TRIG_WHOLE_STEP(c,s) \
    do { c = cos( m_th*radsPerStep - tht ); \
       s = sin( m_th*radsPerStep - tht ); } while(0)
#endif

  //////////////////////////// LINE SETUP //////////////////////////////

  // Test ordering and ensure counterclockwise orientation
  bool ccw = (x1*y2-x2*y1) > 0;
  if (!ccw) {
    // SWAP ENDPOINTS!
    float tmp;
    tmp = x1; x1=x2; x2 = tmp;
    tmp = y1; y1=y2; y2 = tmp;
    tmp = z1; z1=z2; z2 = tmp;
  }

  float dx = x2 - x1;
  float dy = y2 - y1;
  float dz = z2 - z1;
  float r1f = sqrt(x1*x1 + y1*y1);
  float r2f = sqrt(x2*x2 + y2*y2);
  float r1 = half_round( r1f );
  float r2 = half_round( r2f );
  float th1f, th1, th2;
  bool thIgnore = false; // Default is that theta could be dominant
  // Handle degerate endpoint-on-origin case by virtually
  // perturbing origin endpoint towards other endpoint
  if (x1==0 && y1==0) {
    th1f = atan2(y2, x2) * stepsPerRad;
    th1 = th2 = half_round( th1f );
    thIgnore = true;
  }
  else if (x2==0 && y2==0) {
    th1f = atan2(y1, x1) * stepsPerRad;
    th1 = half_round( th1f );
    th2 = th1;
    thIgnore = true;
  }
  else {
    th1f = atan2(y1, x1) * stepsPerRad;
    th1 = half_round( th1f );
    th2 = half_round( atan2(y2, x2) * stepsPerRad );
    if (th1==th2) thIgnore = true;
  }
  float h1 = half_round( z1 );
  float h2 = half_round( z2 );

  // Check to see if we're going around the wrong way.  Fix if so.
  if (th1 > th2)
    th2 += g_stepsA;

  // COMPUTE CHARACTERISTIC LINE CONSTANTS
  // implicit fn F(r,th) = r(Acos(th)+Bsin(th)) = 0
  //const float A = dy;
  //const float B = -dx;
  //const float C = -(A*x1+B*y1);
```

```
const int hSgn = dz < 0 ? -1 : 1;
const float hHalfInc = 0.5 * hSgn;

float polarL2 = dx*dx + dy*dy;  // length squared in 2D polar plane
float polarL2inv, polarLinv;
if (polarL2 < MACHINE_EPSILON) {
  polarL2 = 0;
  polarL2inv = 0;
  polarLinv = 0;
}
else {
  polarL2inv = 1.0/polarL2;
  polarLinv = sqrt(polarL2inv); // inv. len. in 2D polar plane
}

const float rt =                    // radius at point of tangency
  (polarL2==0) ? r1f :(x1*dy-y1*dx)*polarLinv;
const float rt2 = rt*rt;
float ht;                         // height at pt of tangency
// Use endpoint closer to polar origin for less numerical error
if (r1<r2)
  ht = z1 - dz*(x1*dx+y1*dy)*polarL2inv;
else
  ht = z2 - dz*(x2*dx+y2*dy)*polarL2inv;
const float tht = atan2(-dx,dy);     // angle at point of tangency
const float eta = (polarL2inv > 0)? dz*dz*polarL2inv : 2;
// eta = 2 just convenient indication that r never dominates h
const float one_m_eta = 1-eta;
const float xi  = rt * dz*polarLinv;

// must decide the min radius for regions 2- and 1-
float rmin;
bool start_minus;// does first endpoint lie decreasing r part of line?
if (dx*x1 + dy*y1 < 0 && dx*x2 + dy*y2 > 0) {
  rmin = half_round(rt);
  if (rmin<0) rmin = -rmin;
  start_minus = (r1f >= rt);
}
else {
  // minimal radius appears outside the line segment
  rmin = r1<r2 ? r1 : r2;
  start_minus = (r1f >= r2f);
}
// The maximal value for h (premultiplied by the sign of dh)
float hmax = (hSgn*h2<hSgn*ht) ? half_round(hSgn*h2) : half_round(hSgn*ht);

// Because eta_expr is <0 whenever the potential region of h-dominance
// is less than one voxel's h-width, h is never dominant.
const float eta_expr = eta*eta*rt + eta - 1;
const hIgnore = eta_expr < 0 && polarL2 != 0;

// Theta can never be dominant outside of r=0.5 pixels
// if rt < sin(thetastep/2);
if (!thIgnore) {
  thIgnore = (rt < thetaDomLimit);
}

// Left and right hand sides of slope inequality expressions
float dhdth_l, dhdth_r;
float drdh_l;
float drdth_l, drdth_r;
// ... and the sign of drdth_r which is multiplied through.
float drdth_s;

float m_r, m_th, m_h, m_cos, m_sin; // Midpoint test values for variables
float m_hoff;                    // offset from m_h to ht
```

```
  float er_th, er_h;              // Decision variables
  float eh_th_l, eh_th_r, eh_th_s;  // left, right, and right's sign

  float drdth_lInc;               // finite difference increments
  float drdh_lInc, drdh_lIncInc;
  float dhdth_lInc;
  float er_hInc_r, er_hInc_rr, er_hInc_h;

  float r = r1;
  float th = th1;
  float h = h1;

#ifdef INCREMENTAL_TRIG
  float prevCos, prevSin;
#endif

 ////////////////////////// LINE DRAWING //////////////////////////

 /////// FIRST VOXEL ////

 // First voxel is easy when rounding to integer endpoints, but more
 // tricky when using sub-voxel accurate endpoints. Sometimes the voxel
 // the endpoint is actually in should NOT be drawn, because the ideal
 // line would have just nipped the corner of that voxel. What follows
 // is code to determine to back up appropriately so that the first
 // voxel drawn by the standard algorithm is the correct one.
 // All the computation done also gives us what we need to know in order
 // to skip regions of the line drawing code that do not apply.

 m_r = r;
 m_h = h;
 m_th = th;

 // Compute partials (or expression derived from partials and
 // implicit function) at current point to determine dominance.
 m_cos = cos(m_th*radsPerStep - tht);
 m_sin = sin(m_th*radsPerStep - tht);
 if (m_cos < 0) m_cos = 0; // clamp to +- 90 degree range
 drdth_r = m_cos;
 drdth_s = drdth_r < 0 ? -1 : 1;
 drdth_l = m_r * m_sin * radsPerStep * drdth_s;
 drdth_r *= drdth_s;

 dhdth_r = m_cos*m_cos;
 dhdth_l = radsPerStep*xi;

 drdh_l = one_m_eta * m_r*m_r - rt2;

 // determine dominance rankings
 enum { R_DOM_H = 0x4, R_DOM_TH = 0x2, H_DOM_TH = 0x1 };
 unsigned char bits =
//   ((fabs(drdh_l) > fabs(drdh_r) || hIgnore) << 2) | // r dominates h
   ((drdh_l > 0 || hIgnore) << 2) | // r dominates h
   ((fabs(drdth_l) > fabs(drdth_r)) << 1) | // r dominates th
   ((fabs(dhdth_l) > fabs(dhdth_r)) || thIgnore << 0); // h dominates th
 if (th1==th2) bits |= (H_DOM_TH|R_DOM_TH); // can't be TH-dominant

/*
 switch (bits) {
 case 0x1: // 001  r<h r<th h>th => h-th, h-dom
 case 0x0: // 000  r<h r<th h<th => h-th, th-dom
  break;
 case 0x3: // 011  r<h r>th h>th => r-h,  h-dom
 case 0x7: // 111  r>h r>th h>th => r-h,  r-dom
  break;
 case 0x6: // 110  r>h r>th h<th => r-th, r-dom
```

```
   case 0x4: // 100  r>h r<th h<th => r-th, th-dom
    break;
    // case 0x2: // 010  r<h r>th h<th => inconsistent
    // case 0x5: // 101  r>h r<th h>th => inconsistent

  }
 */
  if (r<1)
  {
   // IGNORE WINDUP TO AVOID BACKING UP TO r<0

   // IF ETA BIG, THEN R-DOMINANCE IS IMPOSSIBLE
   if (eta>1.0) bits &= ~(R_DOM_H);

   // IF ACTUAL START POINT IS VERY CLOSE TO R=0, THEN LINE COULD BE
   // TH-DOM EVEN THOUGH IT IS NOT ACCORDING TO SLOPE CHECK AT R=0.5.
   // JUMPING DIRECTLY TO REGION_H_PLUS IS WRONG IN THAT CASE, SO ACT AS IF
   // THETA IS DOMINANT. IF IT REALLY ISN'T TH-DOM, THEN WE'VE ONLY DONE
   // A BIT OF EXTRA WORK.
   if ((bits & H_DOM_TH) && r1f<0.5) bits &= ~(H_DOM_TH);

   setVoxel(r,th,h,COLOR_CYAN);
  }
  else // (r>=1)
  {
   // BASED ON DOMINANCE, BACK UP TO REACH A POINT
   // AT WHICH DRAWING OF THE FIRST VOXEL CAN BE DECIDED
   // LIKE ANY OTHER MID-LINE VOXEL DECISION
   if ((bits & (R_DOM_H|R_DOM_TH)) == 0x0) { //  h-th
    unsigned char hDom = (bits & H_DOM_TH); // h dominant?
    bool belowMidpoint = hDom
     ? hSgn*z1 < hSgn*m_h
     : th1f < m_th;
    if (hDom || belowMidpoint)   h -= hSgn;
    if (m_r != 0) {
     if (!hDom || !belowMidpoint) th -= 1;
    }
   }
  else if ((bits & (R_DOM_TH|H_DOM_TH)) == (R_DOM_TH|H_DOM_TH)) { // r-h
   if (eta>1.0) bits &= ~(R_DOM_H);
   unsigned char rDom = (bits & R_DOM_H); // r dominant?
   int sgn = rDom
    ? start_minus ? -1 : 1
    : (hSgn<0)^start_minus ? -1 : 1;
   bool belowMidpoint = rDom
    ? sgn*r1f < sgn*m_r
    : sgn*z1 < sgn*m_h;
   if (rDom || (belowMidpoint&&polarL2!=0))   r -= start_minus?-1:1;
   if (!rDom || !belowMidpoint) h -= hSgn;
  }
  else if ((bits & (R_DOM_H|H_DOM_TH)) == R_DOM_H
        || (hIgnore&&thIgnore)) {  //  r-th
   unsigned char rDom = (bits & R_DOM_TH); // r dominant?
   int sgn = start_minus ? -1 : 1;
   bool belowMidpoint = (sgn*r1f < sgn*m_r);
   if (rDom || belowMidpoint)    r -= sgn;
   if (!rDom || !belowMidpoint)  th -= 1;
  }

  if (m_th!=th) // we stepped back in theta, recompute sin & cos
  {
   m_th = th;
#ifdef INCREMENTAL_TRIG
   prevCos = m_cos;  prevSin = m_sin;
   // Backwards step -- note change of signs for bW!
   m_cos = prevCos - (aW*prevCos - bW*prevSin);
```

```
        m_sin = prevSin - (aW*prevSin + bW*prevCos);
    #else
        m_cos = cos(m_th*radsPerStep - tht);
        m_sin = sin(m_th*radsPerStep - tht);
    #endif
      }
     } // end if(r>=1)

     // ANY OTHER CASE MUST BE BORDERLINE CASES THAT ARE VERY CLOSE TO BEING
     // DOMINANT IN TWO OR MORE VARIABLES. IN THESE CASES, DO NOT BACKUP.


     // DISTRIBUTOR CAP: JUMP TO APPROPRIATE STARTING REGION
     bits |= ((start_minus&&r1!=0) << 3);
     switch(bits)
     {
      // MUST DO ALL INITIALIZATION EXPECTED BY TARGET REGION BEFORE JUMPING

     case 016: // 1110  r>h r>th h<th => r-th, r-dom
     case 017: // 1111  r>h r>th h>th => r-h,  r-dom
       goto REGION_R_MINUS;
       break;
     case 011: // 1001  r<h r<th h>th => h-th, h-dom
     case 013: // 1011  r<h r>th h>th => r-h,  h-dom
       m_th = th + 0.5;
       TRIG_HALF_STEP(m_cos,m_sin);
       goto REGION_H_MINUS;
       break;
     case 010: // 1000  r<h r<th h<th => h-th, th-dom
     case 014: // 1100  r>h r<th h<th => r-th, th-dom
       m_th = th + 1;
       TRIG_WHOLE_STEP(m_cos,m_sin);
       goto REGION_THETA_MINUS;
       break;
     case 000: // 0000  r<h r<th h<th => h-th, th-dom
     case 004: // 0100  r>h r<th h<th => r-th, th-dom
       m_th = th + 1;
       TRIG_WHOLE_STEP(m_cos,m_sin);
       goto REGION_THETA_PLUS;
       break;
     case 001: // 0001  r<h r<th h>th => h-th, h-dom
     case 003: // 0011  r<h r>th h>th => r-h,  h-dom
       m_th = th + 0.5;
       TRIG_HALF_STEP(m_cos,m_sin);
       goto REGION_H_PLUS;
       break;
     case 006: // 0110  r>h r>th h<th => r-th, r-dom
     case 007: // 0111  r>h r>th h>th => r-h,  r-dom
       m_th = th + 0.5;
       TRIG_HALF_STEP(m_cos,m_sin);
       goto REGION_R_PLUS;
       break;
     }


    /////// REGION R- /////
    REGION_R_MINUS:
    m_r = r-1;
    m_th = th+0.5;
    m_hoff = h+hHalfInc-ht;
    TRIG_HALF_STEP(m_cos,m_sin);

    // The solution in h for the simultaneous equations:
    //      r-h slope == 1  &&  F(r,h) == 0.
    // More robust than just r-h slope test when r-h hyperbola's
    // asymptotes have slope very near +-1 or +-infinity
```

```
  drdh_l = one_m_eta*m_r*m_r - rt2;
  drdh_lInc = one_m_eta*(1-2*m_r);
  drdh_lIncInc = 2*one_m_eta;

  drdth_r = m_cos;
  drdth_s = drdth_r < 0 ? 0 : 1;
  drdth_lInc = m_sin * radsPerStep * drdth_s;
  drdth_l = m_r * drdth_lInc;
  drdth_r *= drdth_s;

  er_th = m_r * m_cos - rt;

  er_h = eta*(m_r*m_r - rt2) - m_hoff*m_hoff;
  er_hInc_r = eta*(1-2*m_r);
  er_hInc_rr = 2*eta;
  er_hInc_h = -hSgn*2*m_hoff - 1;
  // er_hInc_hh = -2;

  while((hIgnore || drdh_l > 0) &&
      (thIgnore || drdth_s==0 || drdth_l < -drdth_r ) &&
      r>rmin
      )
  {
   r -= 1;
   m_r -= 1;

   if (th<th2 && er_th < 0) {
    th += 1;
    m_th += 1;
    TRIG_WHOLE_STEP(m_cos,m_sin);

    drdth_r = m_cos;
    drdth_s = drdth_r < 0 ? 0 : 1;
    drdth_lInc = m_sin * radsPerStep * drdth_s;
    drdth_l = m_r * drdth_lInc;
    drdth_r *= drdth_s;
    er_th = m_r * m_cos - rt;
   }
   else {
    er_th -= m_cos;
   }

   if (h*hSgn<hmax && er_h < 0) {
    h += hSgn;
    drdth_l -= drdth_lInc;
    er_h += er_hInc_h;
    er_hInc_h -= 2; // er_hInc_hh
   }

   drdh_l += drdh_lInc;
   drdh_lInc += drdh_lIncInc;

   er_h += er_hInc_r;
   er_hInc_r += er_hInc_rr;

   setVoxel(r,th,h,COLOR_YELLOW);
  }
  if (r<=rmin && rmin==r2 && th==th2 && h==h2)
   return; // we're done here


//////// REGION H- //////
REGION_H_MINUS:

if (!hIgnore) {
m_r = r - 0.5;
```

```
   m_hoff = h+hSgn-ht;

   drdh_l = m_hoff; // useful in nearly vertical lines to
                    // make sure we don't cross into the PLUS regions

   dhdth_l = m_hoff*m_hoff;
   dhdth_lInc = hSgn*2*m_hoff + 1;
   dhdth_r = xi * (stepsPerRad*hSgn-xi);

   er_h = eta*(m_r*m_r - rt2) - dhdth_l;
   er_hInc_r = eta*(1-2*m_r);
   er_hInc_rr = 2*eta;

   eh_th_l = xi * m_sin;
   eh_th_r = m_cos;
   eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
   eh_th_r *= m_hoff;

   while(hSgn*drdh_l<=0  &&
       (thIgnore ||
        dhdth_l>dhdth_r
        //hSgn*h<dhdth_r
       || xi == 0) &&
       hSgn*h<hmax)
   {
    h += hSgn;
    m_hoff += hSgn;

    if (th<th2 && eh_th_s*eh_th_l < eh_th_s*eh_th_r) {
     th += 1;
     m_th += 1;
     TRIG_WHOLE_STEP(m_cos,m_sin);

     eh_th_l = xi * m_sin;
     eh_th_r = m_cos;
     eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
     eh_th_r *= m_hoff;
    } else {
     eh_th_r += m_cos*hSgn;
    }

    if (er_h > 0 && r>rmin) {
     r -= 1;
     m_r -= 1;
     er_h += er_hInc_r;
     er_hInc_r += er_hInc_rr;
    }

    dhdth_l += dhdth_lInc;
    er_h -= dhdth_lInc;
    dhdth_lInc += 2;

    drdh_l = m_hoff;

    setVoxel(r,th,h,COLOR_GREEN);
  }

 } // end if(!hIgnore)

 //////// REGION THETA- ///

 if (!thIgnore || r<1) { // Can always be theta dominant at polar origin

 m_th += 0.5;
 TRIG_HALF_STEP(m_cos,m_sin);
 REGION_THETA_MINUS:

 m_r = r - 0.5;
 m_hoff = h+hHalfInc-ht;

 drdth_r = m_cos;
 drdth_s = drdth_r < 0 ? 0 : 1; // should never be negative
 drdth_l = m_sin*drdth_s;

 while(drdth_l <= 0 && th<th2)
 {
  eh_th_l = xi * m_sin;
  eh_th_r = m_cos;
  eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
  eh_th_r *= m_hoff;

  er_th = m_r * m_cos - rt;

  if (hSgn*h<h2*hSgn && eh_th_s*eh_th_l > eh_th_s*eh_th_r) {
   h += hSgn;
   m_hoff += hSgn;
  }
  if (er_th > 0 && r>rmin) {
   r -= 1;
   m_r -= 1;
  }

  th += 1;
  m_th += 1;
  TRIG_WHOLE_STEP(m_cos,m_sin);

  drdth_r = m_cos;
  drdth_s = drdth_r < 0 ? 0 : 1; // should never be negative
  drdth_l = m_sin*drdth_s;

  setVoxel(r,th,h,COLOR_MAGENTA);
}

/////// REGION THETA+ ///
REGION_THETA_PLUS:

m_r = r + 0.5;
m_hoff = h + hHalfInc - ht;

dhdth_l = radsPerStep*xi;
dhdth_r = m_cos*m_cos;

// Implicit fn of just theta. Derived from:
//    dr/dth==1 and F(r,th) == 0
// More robust than plain dr/dth check for lines close to polar origin.
drdth_l = dhdth_r - rt*m_sin*radsPerStep;

while(drdth_l>0 &&
   hSgn*dhdth_l < dhdth_r &&
   th<th2 &&
   m_cos > 0)
{
 eh_th_l = xi * m_sin;
 eh_th_r = m_cos;
 eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
 eh_th_r *= m_hoff;

 er_th = m_r * m_cos - rt;

 if (hSgn*h<h2*hSgn &&
    eh_th_s*eh_th_l > eh_th_s*eh_th_r) {
     h += hSgn;
     m_hoff += hSgn;
    }
    if (r<r2 && er_th < 0) {
     r += 1;
     m_r += 1;
    }
    th += 1;
    m_th += 1;
    TRIG_WHOLE_STEP(m_cos,m_sin);

    dhdth_r = m_cos*m_cos;
    drdth_l = dhdth_r - rt*m_sin*radsPerStep;

    setVoxel(r,th,h,COLOR_RED);
   }
  } // end !thIgnore

  //////// REGION H+ //////

  m_th = th + 0.5;
 #ifdef INCREMENTAL_TRIG
  // this is a half step from the PREVIOUS cos and sin values
  m_cos = prevCos - (aH*prevCos + bH*prevSin);
  m_sin = prevSin - (aH*prevSin - bH*prevCos);
 #else
  m_cos = cos( m_th*radsPerStep - tht );
  m_sin = sin( m_th*radsPerStep - tht );
 #endif

  REGION_H_PLUS:

  m_r = r + 0.5;
  m_hoff = h+hSgn - ht;

  // The solution in h for the simultaneous equations:
  //      r-h slope == 1  &&  F(r,h) == 0.
  // More robust than just r-h slope test when r-h hyperbola's
  // asymptotes have slope very near +/-1 or +/-infinity
  drdh_l = one_m_eta*m_r*m_r - rt2;
  drdh_lInc = one_m_eta*(1-2*m_r);
  drdh_lIncInc = 2*one_m_eta;

  eh_th_l = xi * m_sin;
  eh_th_r = m_cos;
  eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
  eh_th_r *= m_hoff;

  er_h = eta*(m_r*m_r - rt2) - m_hoff*m_hoff;
  er_hInc_r = eta*(1+2*m_r);
  er_hInc_rr = 2*eta;
  er_hInc_h = -hSgn*2*m_hoff - 1;

  while(drdh_l < 0 || eta > 1.0)
  {
   if (h*hSgn >= h2*hSgn) return; // gone far enough--skip region r+;

   h += hSgn;
   m_hoff += hSgn;

   if (th<th2 && eh_th_s*eh_th_l < eh_th_s*eh_th_r) {
    th += 1;
    m_th += 1;
    TRIG_WHOLE_STEP(m_cos, m_sin);
    eh_th_l = xi * m_sin;
    eh_th_r = m_cos;
```

```
      eh_th_s = hSgn*(eh_th_r < 0 ? -1 : 1);
      eh_th_r *= m_hoff;
     }
     else {
      eh_th_r += m_cos*hSgn;
     }

     if (r<r2 && er_h < 0) {
      r += 1;
      m_r += 1;
      drdh_l += drdh_lInc;
      drdh_lInc += drdh_lIncInc;
      er_h += er_hInc_r;
      er_hInc_r += er_hInc_rr;
     }

    er_h += er_hInc_h;
    er_hInc_h -= 2;

    setVoxel(r,th,h,COLOR_GREEN);
   }


  //////// REGION R+ //////
  REGION_R_PLUS:
  m_r = r+1;
  m_hoff = h+hHalfInc-ht;

  er_th = m_r * m_cos - rt;

  er_h = eta*(m_r*m_r - rt2) - m_hoff*m_hoff;
  er_hInc_r = eta*(1+2*m_r);
  er_hInc_rr = 2*eta;
  er_hInc_h = -hSgn*2*m_hoff - 1;

  while(r<r2 && h*hSgn <= h2*hSgn && th<=th2)
  {
   r += 1;
   m_r += 1;

   if (th<th2 && er_th > 0) {
     th += 1;
     m_th += 1;
#ifdef INCREMENTAL_TRIG
     TRIG_WHOLE_STEP(m_cos,m_sin);
#else
     m_cos = cos( m_th*radsPerStep - tht );
#endif
     er_th = m_r * m_cos - rt;
   }
   else {
     er_th += m_cos;
   }

   if (hSgn*h < hSgn*h2 && er_h > 0) {
     h += hSgn;
     er_h += er_hInc_h;
     er_hInc_h -= 2;
   }

   er_h += er_hInc_r;
   er_hInc_r += er_hInc_rr;

   setVoxel(r,th,h,COLOR_YELLOW);
  }
}
```

Having described the invention and a prederred embodiment thereof, we claim:

CLAIMS

1. A method for rendering, on a volumetric display having a plurality of voxels defined by cylindrical coordinates, a rasterized line that approximates a desired line, said method comprising:

   rendering, on said volumetric display, a first voxel from said plurality of voxels, the first voxel being representative of a first point from said desired line, said first point being selected from a first constituent line-segment of said desired line, said first constituent line-segment having a first dominant direction;

   selecting a second point from said desired line;

   determining whether said second point is on a second constituent line-segment of said desired line, said second constituent line-segment having a second dominant direction different from said first dominant direction;

   on the basis of said determination, rendering a second voxel from said plurality of voxels, said second voxel being representative of said second point on said desired lineand having a location that minimizes a distance between said second voxel and said second point on said desired line.

2. The method of claim **1** wherein determining whether said second point is on a second constituent line-segment of said desired line comprises

   evaluating a difference between a threshold and a transition-indicating expression evaluated at said second point, said difference being indicative of a transition from said first constituent line-segment to said second constituent line-segment.

3. The method of claim **2**, further comprising determining said second dominant direction.

4. The method of claim 3, wherein determining said second dominant direction comprises selecting said second dominant direction from a subsequent direction in a pre-defined progression of dominant directions.

5. The method of claim 4, further comprising defining said progression of dominant directions to be a sequence consisting of:

a radial direction followed by

an axial direction followed by

an angular direction, followed by

said axial direction, followed by

said radial direction.

6. The method of claim 2, further comprising selecting said transition indicating expression to include an expression indicative of a slope of said desired line at said second point.

7. The method of claim 6, wherein said first dominant direction defines first and second non-dominant directions and said slope is defined to be a ratio of how much said desired line changes in said first dominant direction with respect to a change in one of said first and second non-dominant directions.

8. The method of claim 6, wherein said transition indicating expression is selected to identify a line having a point at which said slope is equal to said threshold.

9. The method of claim 1, wherein rendering said second voxel comprises applying a midpoint method to select said second voxel from a plurality of candidate voxels.

10. A method of rendering, on a display having a plurality of voxels defined by a cylindrical coordinate system, a rasterized line that approximates a desired line, the method comprising:

rendering a first voxel on said rasterized line, said first voxel being representative of a first point on said desired line;

determining, on the basis of a local slope of said desired line and a sign of a discriminant of said desired line in a neighborhood of said first point, a direction relative to said first voxel of a second voxel on said rasterized line;

rendering said second voxel on said rasterized line, said second voxel being displaced from said first voxel in said direction.

11. The method of claim **10** further comprising evaluating said discriminant at a midpoint between two candidate voxels from which said subsequent voxel is to be selected.

12. The method of claim **10** wherein determining a direction comprises partitioning said desired line into a plurality of constituent line-segments, each constituent line-segment being associated with a range of local slopes of said desired line.

13. The method of claim **12** further comprising updating said discriminant.

14. The method of claim **12** wherein updating said discriminant comprises:

incrementing said discriminant by a selected amount when said subsequent voxel is displaced from said first voxel in a first direction; and

incorporating information indicative of a displacement of said second voxel in a second direction.

15. A volumetric display comprising:

a display surface rotatable about an axis;

an voxel-rendering element configured to render a selected voxel on said display surface; and

a rasterizer in communication with said voxel-rendering element, said rasterizer being configured to direct said voxel-rendering elemen to render, on said display surface, a rasterized line that approximates a desired line by:

rendering, on said display surface, a first voxel defined by first cylindrical coordinates, said first voxel being representative of a first point from said desired line, said first point being selected from a first constituent line-segment of said desired line, said first constituent line-segment having a first dominant direction;

selecting a second point from said desired line;

determining whether said second point is on a second constituent line-segment of said desired line, said second constituent line-segment having a second dominant direction different from said first dominant direction;

on the basis of said determination, rendering a second voxel representative of said second point on said desired line, said second voxel having a location defined by second cylindrical coordinates that minimizes a distance between said second voxel and said desired line.

A method for rendering, on a volumetric display having a plurality of voxels defined by cylindrical coordinates, a rasterized line that approximates a desired line, said method comprising:

determining whether first and second points on the desired line are on first and second constituent line-segments of said desired line, said first and second constituent line-segments having different dominant directions;

on the basis of said determination, incrementally proceeding from a first voxel representative of said first point to a second second voxel representative of said second point on said desired line,

rendering said second voxel at a location defined by cylindrical coordinates that minimizes a distance between said second voxel and said desired line.

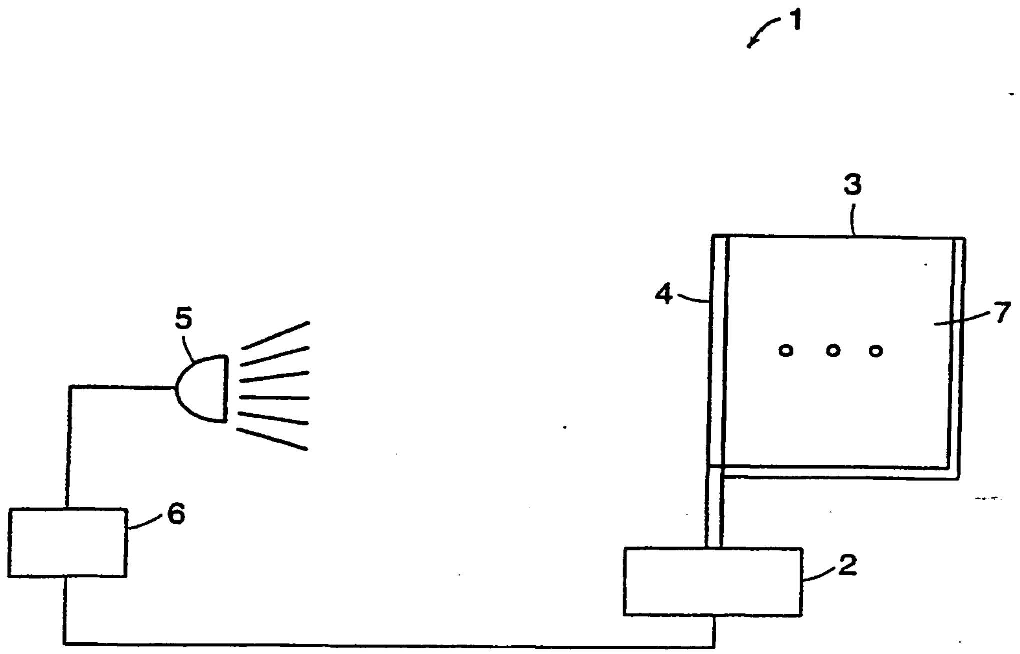


FIG. 1

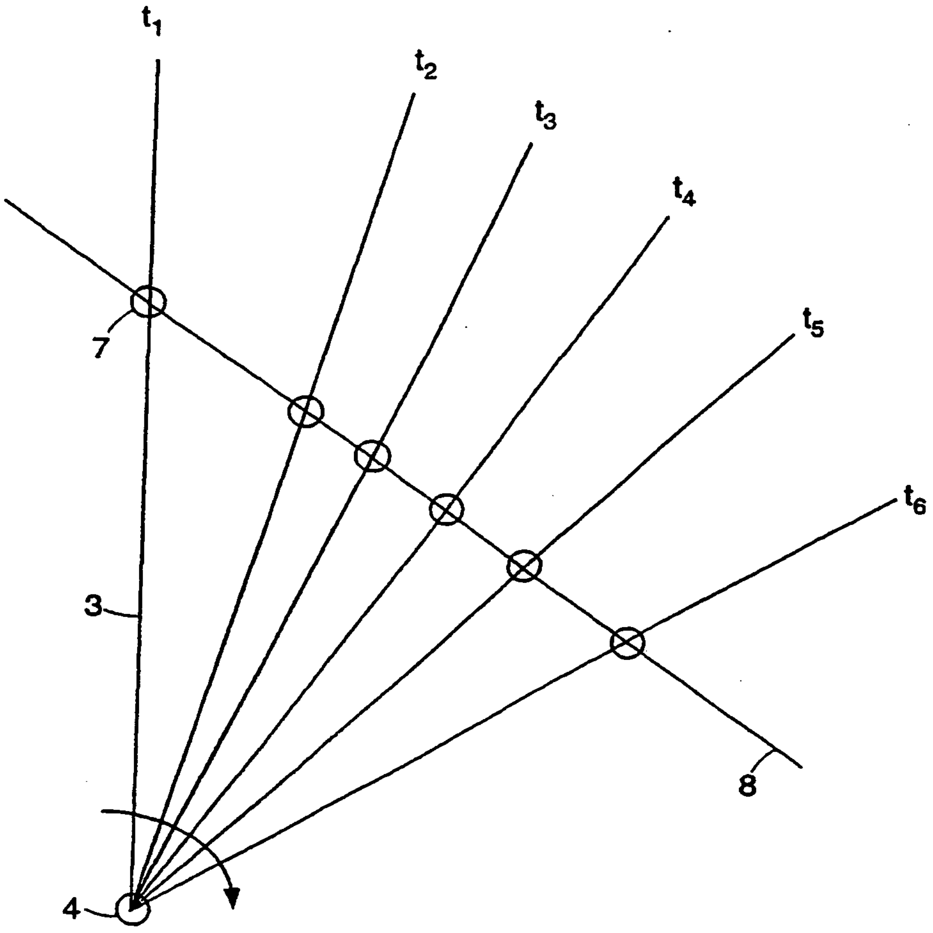


FIG. 2

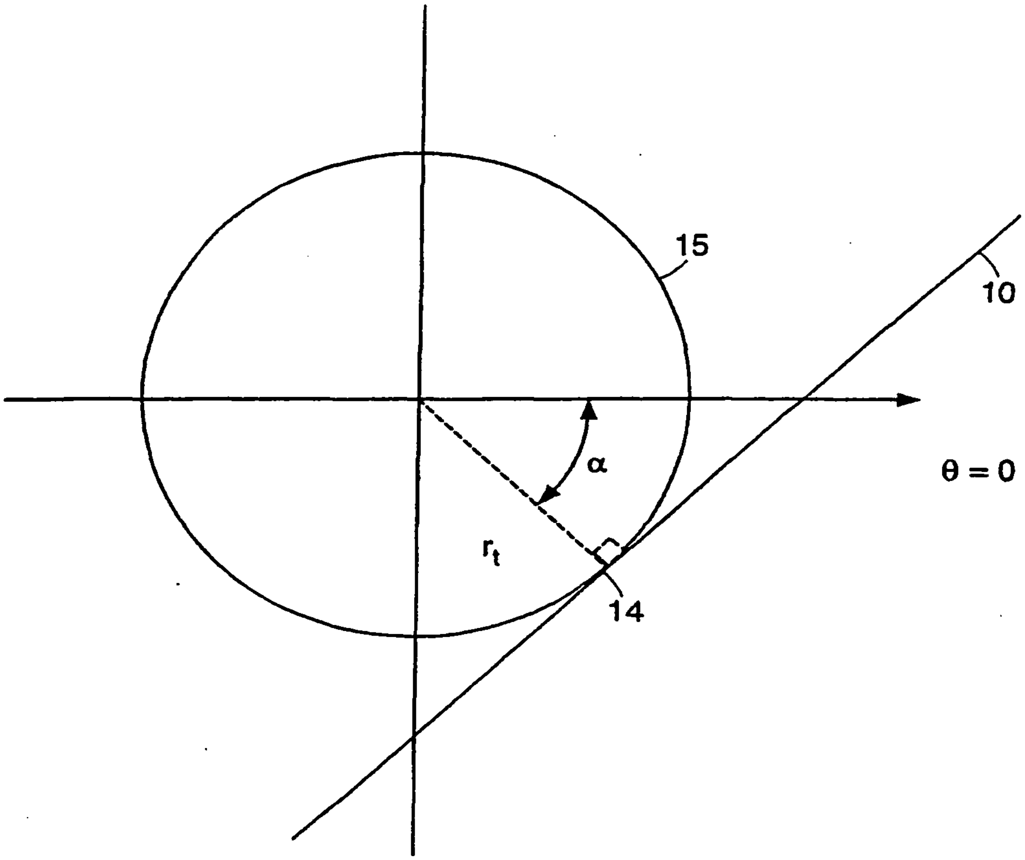


FIG. 3

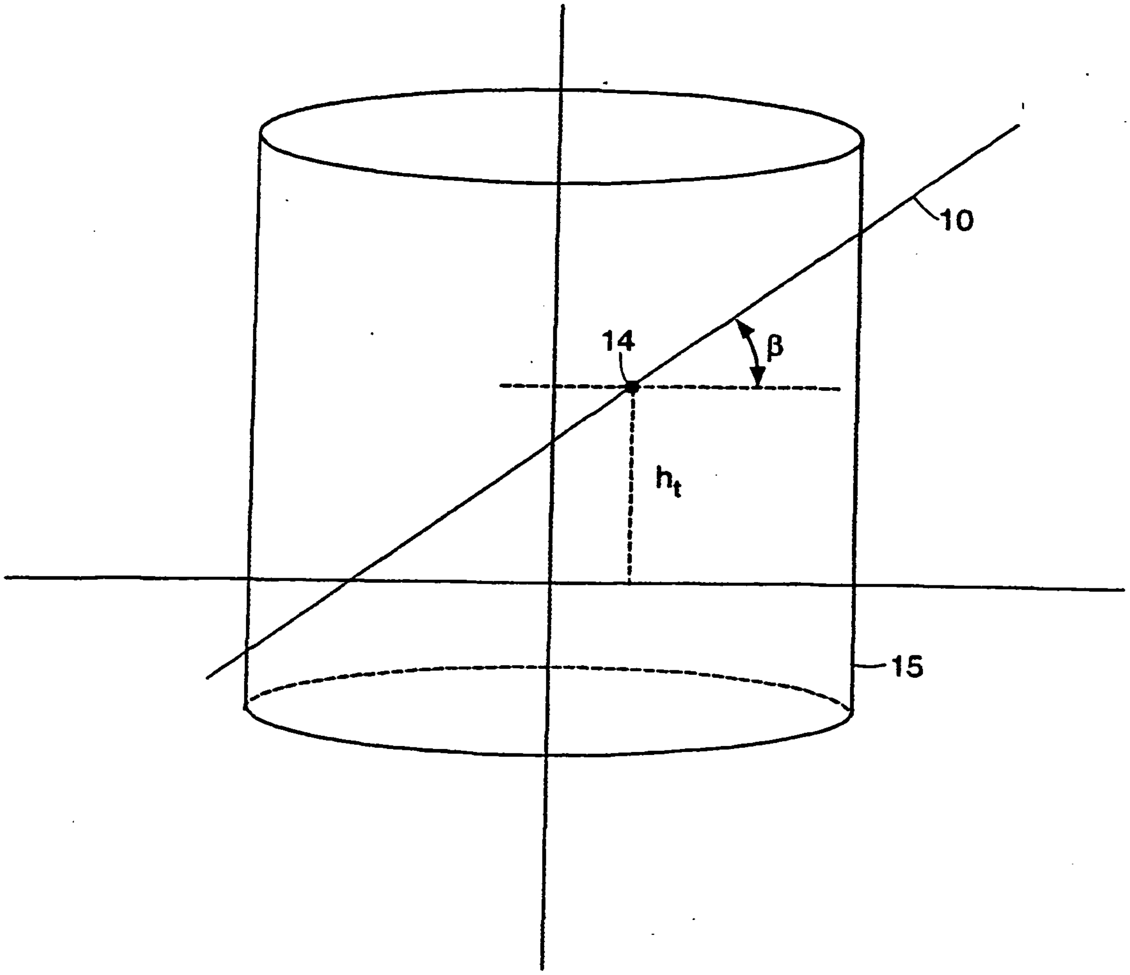


FIG. 4

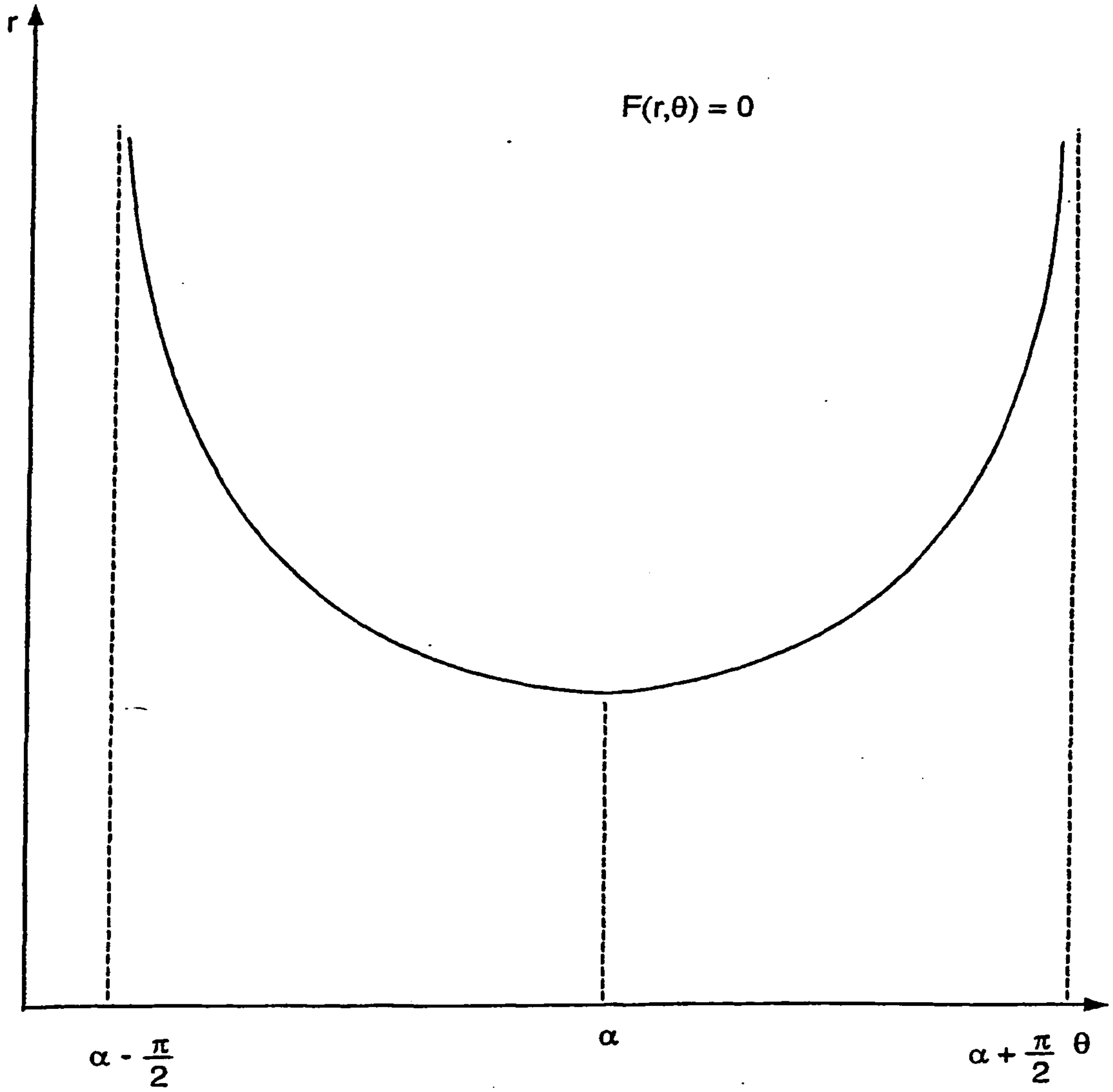


FIG. 5

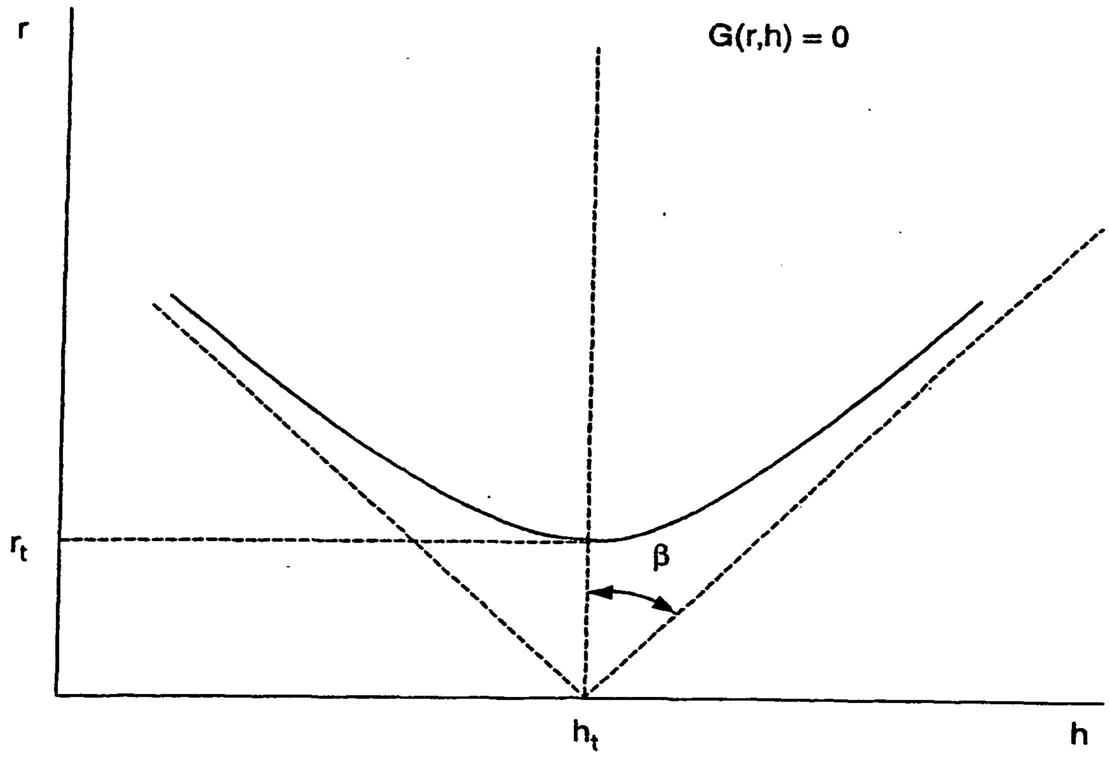


FIG. 6

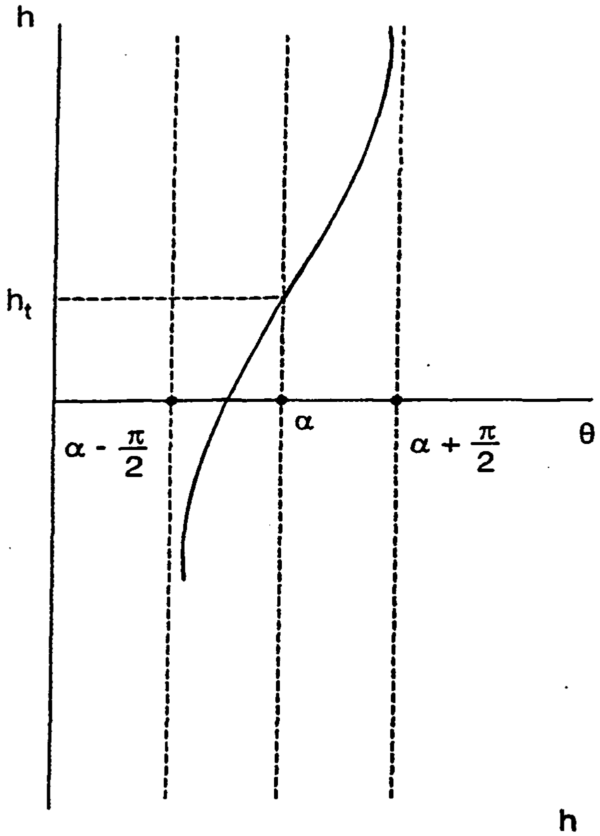

h
h_t
$\alpha - \frac{\pi}{2}$
α
$\alpha + \frac{\pi}{2}$
θ

FIG. 7A

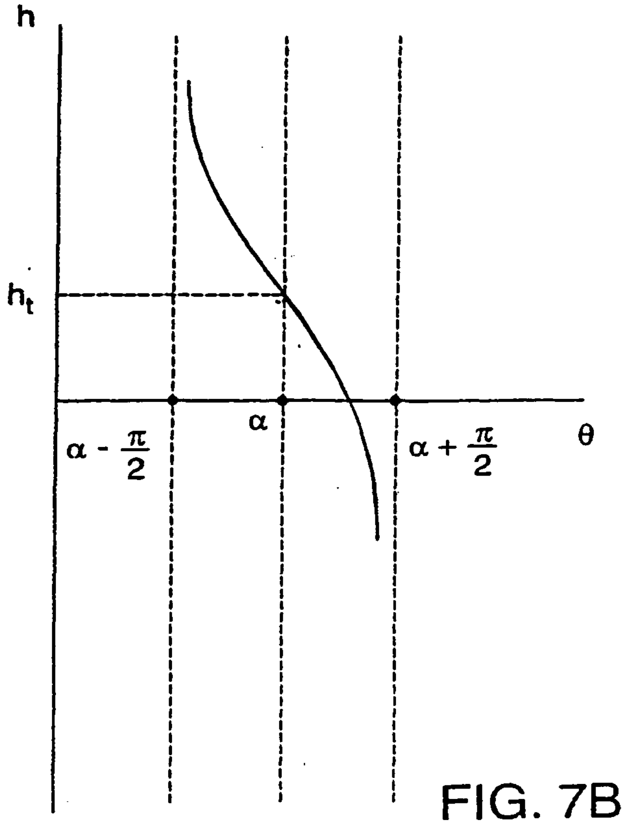

h
h_t
$\alpha - \frac{\pi}{2}$
α
$\alpha + \frac{\pi}{2}$
θ

FIG. 7B

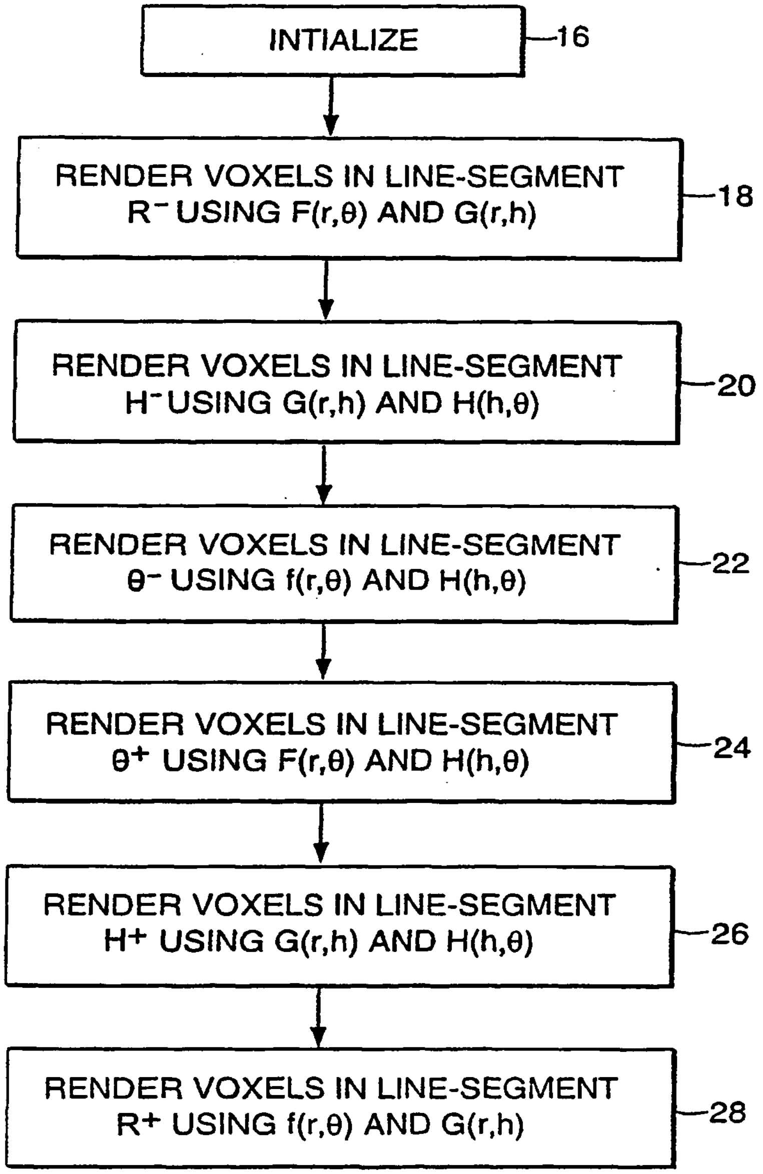

INTIALIZE
16
RENDER VOXELS IN LINE-SEGMENT
R^- USING $F(r,\theta)$ AND $G(r,h)$
18
RENDER VOXELS IN LINE-SEGMENT
H^- USING $G(r,h)$ AND $H(h,\theta)$
20
RENDER VOXELS IN LINE-SEGMENT
θ^- USING $f(r,\theta)$ AND $H(h,\theta)$
22
RENDER VOXELS IN LINE-SEGMENT
θ^+ USING $F(r,\theta)$ AND $H(h,\theta)$
24
RENDER VOXELS IN LINE-SEGMENT
H^+ USING $G(r,h)$ AND $H(h,\theta)$
26
RENDER VOXELS IN LINE-SEGMENT
R^+ USING $f(r,\theta)$ AND $G(r,h)$
28

FIG. 8

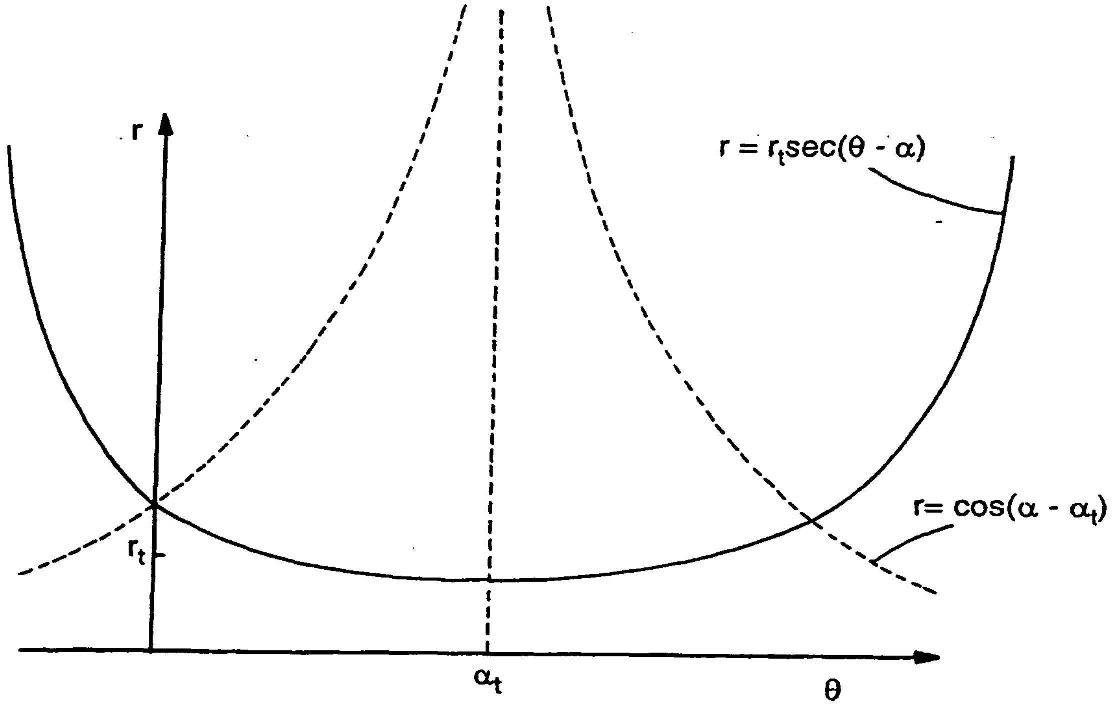


FIG. 9

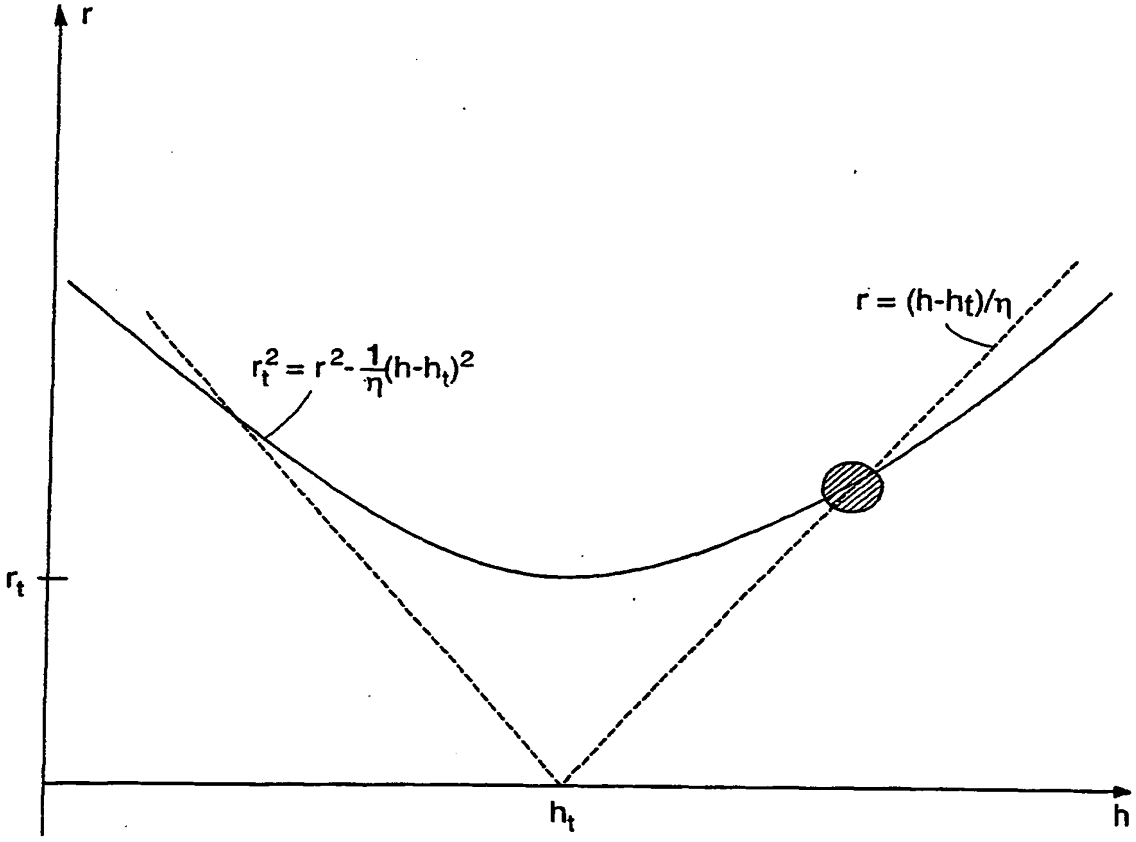


FIG. 10

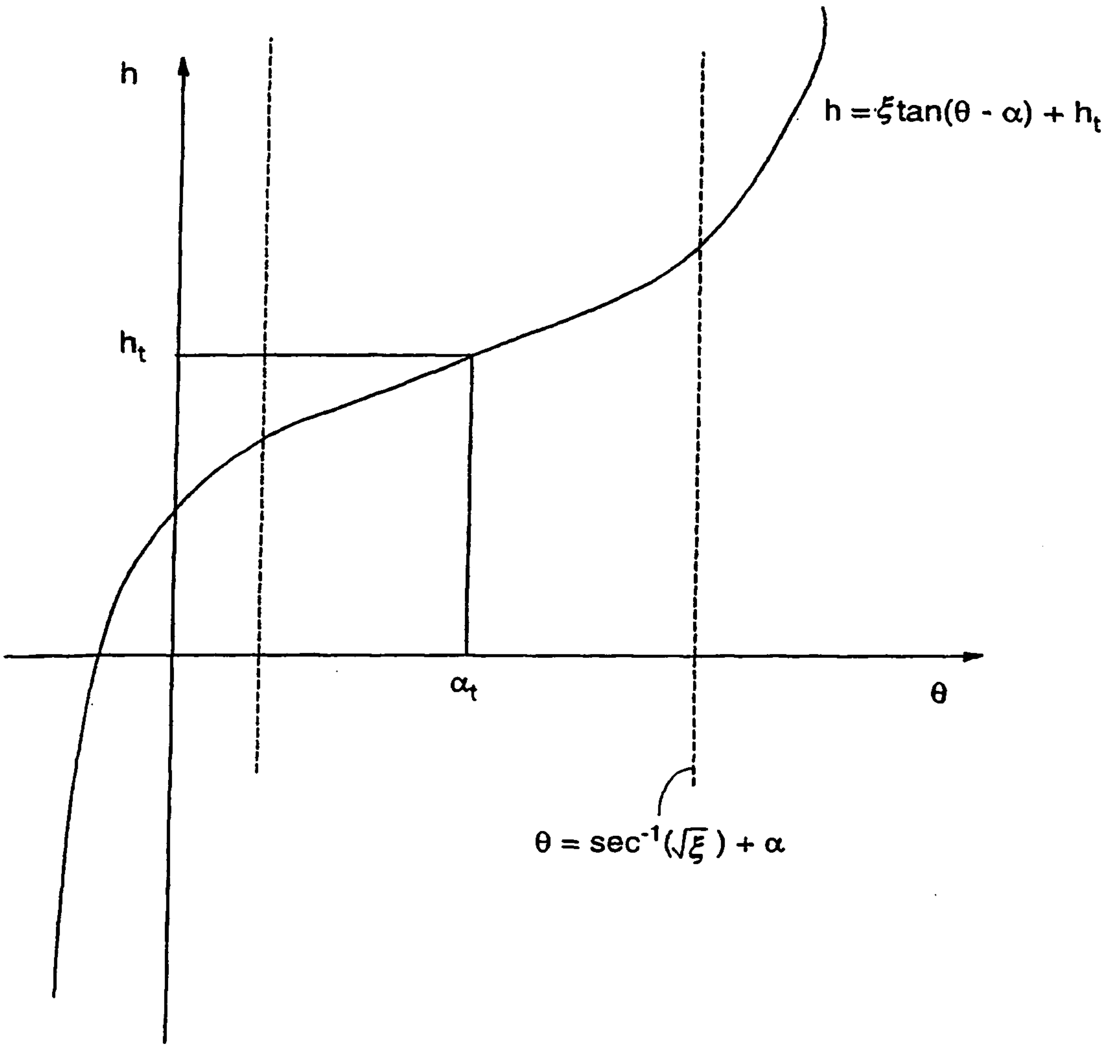

h
$h = \xi \tan(\theta - \alpha) + h_t$
h_t
α_t
θ
$\theta = \sec^{-1}(\sqrt{\xi}\,) + \alpha$

FIG. 11

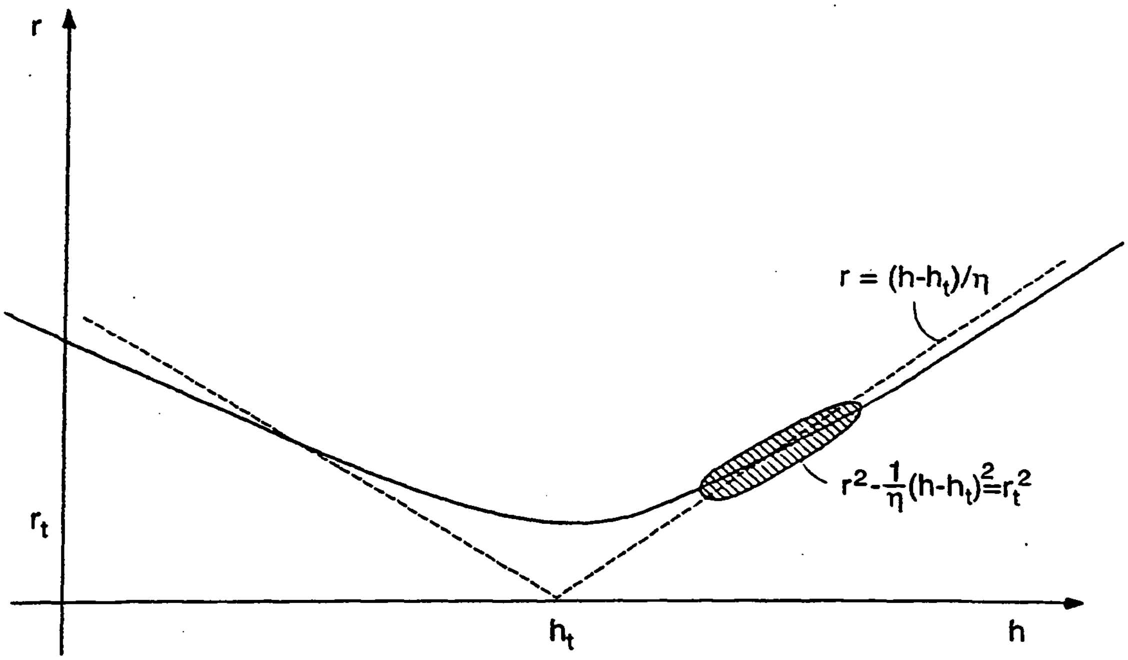


FIG. 12

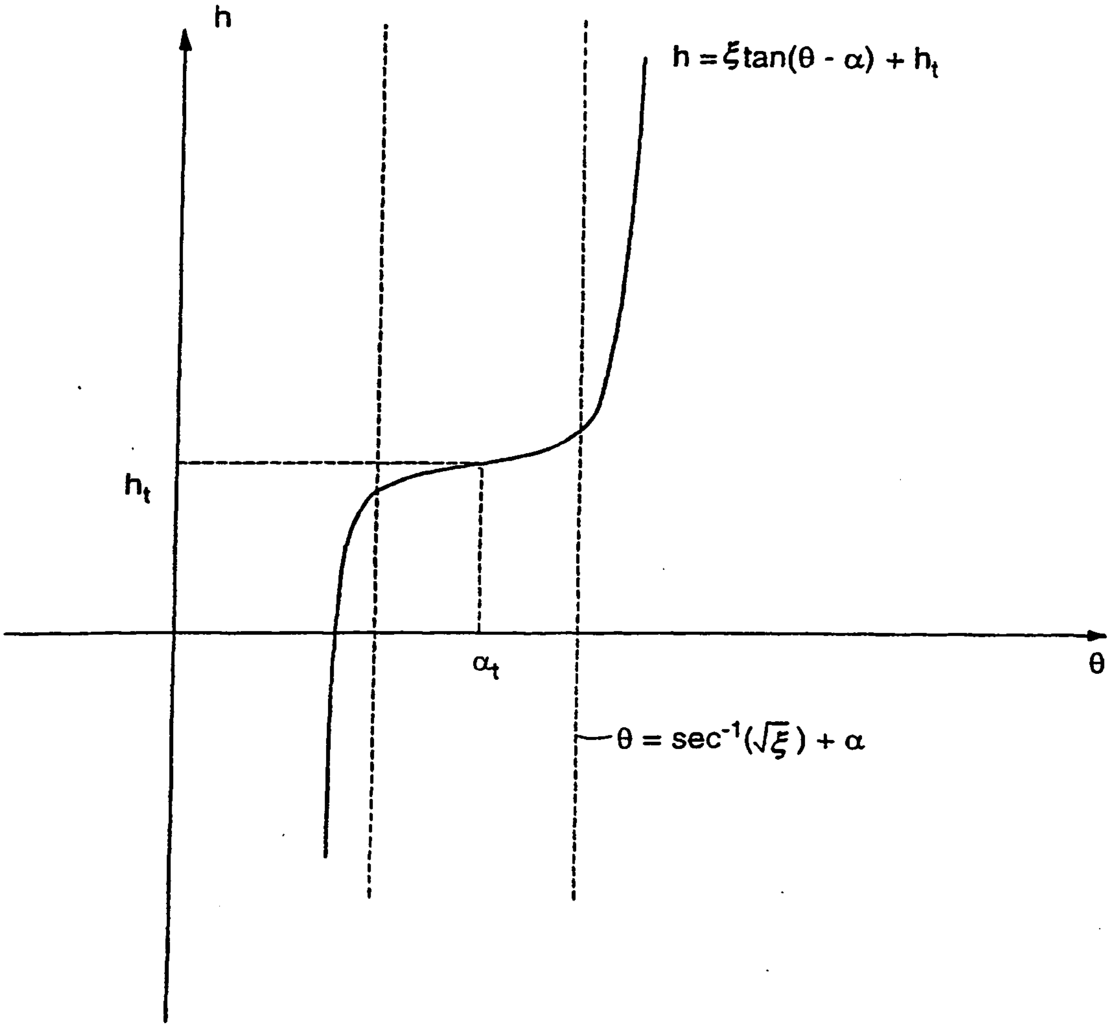

h
$h = \xi \tan(\theta - \alpha) + h_t$
h_t
α_t
θ
$\theta = \sec^{-1}(\sqrt{\xi}) + \alpha$

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/32246

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06T 17/00
US CL :345/424
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (*classification system followed by classification symbols*)

U.S. : 345/6,419,422,424

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Foley, *Computer Graphics: Principles and Practice*

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,515,484 A (SFARTI et al) 07 May 1996 col 5 lines 25 -35, col 9 lines 19 - 21	1-16
Y	US 5,754,147 A (TSAO et al.) 19 May 1998, figs. 9a-9c	1-16
Y	FOLEY J. ET AL. Computer Graphics Principles and Practice Addison-Wesley Publishing Co. July 1995. 2nd ed. pages 74-78 under title "Mid-Point Line Algorithm"	9,11

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

*	*Special categories of cited documents:*	"T"	*later document published after the international filing date or priority* date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	*document of particular relevance;* the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date		
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document *published prior to the international filing date but later* than the priority date claimed	"&"	*document member of the same patent family*

Date of the actual completion of the international search

15 JANUARY 2002

Date of mailing of the international search report

12 FEB 2002

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer for
SAM A WILEY Eugenia Logan
Telephone No. (703) 605 - 4248

Form PCT/ISA/210 (second sheet) (July 1998)★